

<b>Modul INF-MSc-331: Funktionallogisches Modellieren und Programmieren (FLMP)</b>					
<b>Deutscher Modultitel:</b> Functional-logic modeling and programming					
<b>Studiengänge:</b> Masterstudiengang Informatik, Masterstudiengang Angewandte Informatik					
<b>Turnus</b> nach Ankündigung	<b>Dauer</b> 1 Semester	<b>Studienabschnitt</b> 2.-3. Semester	<b>Credits</b> 6	<b>Aufwand</b> 180 (60/120)	
<b>1</b>	<b>Modulstruktur</b>				
	<b>Nr.</b>	<b>Element / Lehrveranstaltung</b>	<b>Typ</b>	<b>Credits</b>	<b>SWS</b>
	1	Funktionallogisches Modellieren und Programmieren	V	3	2
	2	Übung zu Funktionallogisches Modellieren und Programmieren	Ü o. Proj.	3	2
<b>2</b>	<b>Lehrveranstaltungssprache:</b> deutsch oder englisch				
<b>3</b>	<b>Lehrinhalte</b> Die Lehrveranstaltung behandelt grundlegende Konzepte zu Konstruktion, Ausführung und Verifikation formaler Modelle. 50 Jahre Forschung und Entwicklung an der Schnittstelle zwischen Mathematik und Softwaretechnik haben gezeigt, dass Aufbau und Verhalten der Objekte eines formalen Modells von spezifischen Konstruktoren bzw. Destruktoren bestimmt werden. Konstruktormengen werden induktiv als Grammatiken o.ä. definiert. Destruktormengen bestehen aus Transitions- und Attributfunktionen zustandsbasierter Systeme (Automaten, Kripke-Strukturen, Petri-Netze, Flussgraphen, Prozessalgebren, Klassenhierarchien, etc.) und werden heute oft coinduktiv definiert. Beide Modelltypen verfügen über mächtige Auswertungs-, Lösungs- und Beweismethoden. Die LV wird zunächst fundamentale typ- und kategorientheoretische Konzepte (Funktoren, Limiten, Algebren, Folds, Unfolds, Monaden) behandeln und darauf aufbauend eine an Haskell, Prolog und SQL angelehnte funktionallogische Sprache, die im interaktiven Spezifikationswerkzeug Expander2/3 ausführbar ist.				
<b>4</b>	<b>Kompetenzen</b> Die Studierenden lernen, welche grundlegenden mathematischen Konstrukte inkl. Entwurfsmuster und Beweisverfahren für welche Anwendungen geeignet sind und wie sie an dortige Anforderungen angepasst werden können. Anstatt gängige formale Methoden getrennt voneinander zu betrachten, werden alle behandelten Anwendungen in einer einzigen, aber mächtigen und flexiblen logisch-algebraischen Sprache formuliert. So wird der souveräne Umgang mit logisch-algebraischen Techniken gefördert, der nicht nur bei der Synthese oder Analyse von Programmen weiterhilft, sondern auch bei der Auswahl für die jeweilige Anwendung geeigneter Entwurfs- und Programmierwerkzeuge.				
<b>5</b>	<b>Prüfungen</b> <i>Modulprüfung:</i> mündliche Prüfung (30 Minuten) <sup>BOSS-NR. ?????</sup> <i>Studienleistung:</i> <ul style="list-style-type: none"> <li>Aktive Teilnahme in den Übungen</li> <li>Erreichen einer Mindestpunktzahl der Übungsaufgaben <sup>BOSS-NR. ?????</sup></li> </ul> Die Studienleistung ist Voraussetzung für die Teilnahme an der Modulprüfung.				
<b>6</b>	<b>Prüfungsformen und -leistungen</b> <input checked="" type="checkbox"/> Modulprüfung <input type="checkbox"/> Teilleistungen				
<b>7</b>	<b>Teilnahmevoraussetzungen</b> <i>Erfolgreich abgeschlossen:</i> Ein Basismodul aus einem beliebigen Forschungsbereich <i>Wünschenswerte Kenntnisse:</i> Funktionale und/oder logische Programmierung, mathematische Grundbegriffe				
<b>8</b>	<b>Modultyp und Verwendbarkeit des Moduls</b> Vertiefungsmodul in den Masterstudiengängen Informatik und Angewandte Informatik Forschungsbereich: Software, Sicherheit und Verifikation Das Modul kann nicht gewählt werden, wenn mindestens eines der beiden Vertiefungsmodule MSc-304 „Funktionales und regelbasiertes Programmieren“ und MSc-318 „Logisch-algebraischer				

	Systementwurf“ bestanden wurde.		
<b>9</b>	<b>Modulbeauftragte/r</b> Prof. Dr. P. Padawitz	<b>Zuständige Fakultät</b> Informatik	Beschluss Fakultätsrat 18.10.2022